This is a summary of new features added to PDP-11 BASIC-PLUS-2 for Version 2.3:

- o BASIC now provides I-space and D-Space support. Using I-space and D-space can provide twice as much virtual address space for user tasks. Contact your system manager to see if your hardware and software support I-space and D-space. For more information on I-space and D-space support, please see the BASIC on RSX-11M/M-PLUS Systems manual.

- o RESUME to <Label> is now allowed. In previous versions of BASIC-PLUS-2, the target of a RESUME statement was a line number. In BASIC-PLUS-2 V2.3, you can specify a label as the target of a RESUME statement.

- o Larger programs can be compiled. In previous versions of BASIC-PLUS-2, compiling very large programs could cause the compiler to fail with a "work file error #12". In BASIC-PLUS-2 V2.3, the size of the compiler work files has increased to allow larger programs to be compiled.

- o Enhancements have been added to the compiler to improve the speed at compilation. The improvements do not require using a secondary cache; however, if you use a secondary cache, performance is enhanced even more.

- o You do not have to recompile programs for BASIC-PLUS-2 V2.3. However, if your programs link to the BASIC-PLUS-2 resident library, you must relink your programs to the new resident library. Also, programs compiled with BASIC-PLUS-2 V2.3 cannot execute properly if linked to the BASIC-PLUS-2 OTS V2.2.

See the release notes for more complete information. Many bugs were also cleaned up.

The following notes are additional release notes that were not available when the installation guide and release notes for your system went to print.

- If the /SYNTAX_CHECKING qualifier is set, entering a SUB or FUNCTION statement in the BASIC environment causes the compiler to hang. Typing a CTRL/C clears the condition. To avoid this situation, do not set the /SYNTAX qualifier until the SUB or FUNCTION statement has been entered.

- Compiling a program that consists of only a SUB or FUNCTION statement results in an internal error. To correct this error, and complete the program logic, include an END SUB or END FUNCTION statement.

- In the BASIC environment, using the RUN command twice without altering the program code or the default compiler settings, results in the same task image being run both times. As a result, qualifiers added to the second RUN command are ignored. For example, if you first run the task with the RUN/WORD command, and then run it with the RUN/BYTE command, the default data type for both runs is WORD.

  If you do not alter the program code, you can avoid this problem by resetting the default compiler settings with the SET command. That is, use the commands RUN/WORD, SET/BYTE, and RUN. Now the second execution has the default data type BYTE.

- Programs linked to the large resident library run slower than programs linked to the small resident library or programs not linked to a resident library. However, large programs may need to be overlaid if not linked with the large resident library, causing slower execution speed.

- BYTE expressions that normally overflow may cause bad results. For example:

```
A% = 100%
PRINT A% + A%
```

  If A is a BYTE integer, the expression in the PRINT statement produces a result of 200, instead of a negative BYTE value. To avoid this situation, assign the BYTE expression to a BYTE variable. For example:

```
A% = 100%
B% = A% + A%
PRINT B%
```

- If a program contains a BYTE value or expression multiply assigned to BYTE array elements, running it in the BASIC environment results in an "odd address trap" error. For example:

```
10 DECLARE BYTE A(5), B(5), C
   A(4%), B(4%) = C
```

Compiling and linking the program does not produce this error.

- While running a program in the BASIC environment, you may receive the run-time error "no room for user on device" (error number 4). This can occur because BASIC tries to store the task as a contiguous file and there is not enough contiguous room on the disk. If this is the case, you may be able to solve the problem by compressing the disk using the DSC utility.

****

B P 2 B . D O C

****

Before you update BASIC-PLUS-2, you must copy the file DIALOG.TSK from the update directory to the account from which you plan to do the installation. When installing BASIC-PLUS-2, the RSTS/E BUILD utility will prompt you for the name of a control file. When reponding to the "Control file is ?" prompt, you must specify the update account in addition to "BP2". For example:

```
Control file is ?  SY:[200,200]BP2
```

Follow the instructions in your installation guide, to complete the BASIC-PLUS-2 installation.

****

The following changes have been made to the BASIC-PLUS-2 installation which are not documented in the Release Notes:

- o The installation procedure now works correctly on non-FPU machines.
- o BP2BLD.TSK now correctly modifies the PRINT USING statement should you choose non-default characters for the currency symbol, decimal point, or third digit seperator.
- o The following table contains a list of all changes made to BASIC-PLUS-2 V2.3 for RSTS/E V9.3. An explanation is provided which describes the reason for the update.

Module	Library	Explanation
STLSS	BP2EOT,BP2FOT	Corrects LSET when used with the null string.
STCFS	BP2EOT,BP2FOT	Corrects usage of the null string in the CVT$% function.
STFN1	BP2EOT,BP2FOT	Corrects usage of the null string in the ASCII function.
CDGEN1	RSTLIB	Corrects division of BYTE expressions. Now optimizes CMP$ to TST$ when one operand is zero. Corrects the concatenation of string array elements declared within a dynamic MAP. Corrects the indexed GET with a KEY expression other than an type INTEGER.
CDATA	RSTLIB	Corrects subtraction of BYTE expressions. Incorporates the optimization made in CDGEN1.
OBJGSD	RSTLIB	Eliminates an erroneous error message when declaring arrays in MAPS/COMMONS.
OUTMAL	RSTLIB	Eliminates an erroneous error message when declaring arrays in MAPS/COMMONS.
RUNALC	RSTLIB	Eliminates an erroneous error message when declaring arrays in MAPS/COMMONS.

SARLOO RSXCFELIB Corrects ITERATE in FOR/WHILE loops.